

**REMARKS**

In view of the foregoing amendment and following remarks responsive to the Office Action dated December 19, 2005, Applicant respectfully requests favorable reconsideration of this application.

Applicant has herein slightly amended various claims in order to improve their form. More particularly, all of the changes in the claims except for one of the changes in claim 1 make the language more internally consistent in accordance with the language convention chosen by applicant in the claims. The aforementioned one additional change in claim 1, is the addition of the recitation that the order data comprises data other than the order in which the references to the supplemental files appear in the Web page.

This case had previously been appealed and an appeal brief was filed on February 3, 2005. The latest Office Action reopens prosecution and asserts a new ground of rejection. More specifically, all claims, claims 1-22, had previously been rejected as anticipated by Anders. Under the new rejection, all claims, claims 1-22, are rejected as obvious over Anders in view of Krishnan.

Applicant respectfully traverses.

**The Present Invention**

The present invention is a method and apparatus for loading web pages, including supplemental files such as pictures, sound files, video files, etc., at a browser.

One of the problems of the prior art is that browsers read the HTML code in a Web page from left to right and from top to bottom. Accordingly, the browser encounters the embedded references to supplemental files in the order in which they are encountered while reading the page. The browser will send requests back to the server for those supplemental files in the order that the browser encounters the references while reading the HTML code. Since a browser has a limited number of ports, the supplemental files may not be retrieved and loaded in the most efficient manner. For instance, if a browser has four ports and the requested page has 14 supplemental files, in which the first four referenced supplemental files are large files and the next 10 are small files, the browser may take a long time to download the first four large files, while the person sitting at the client browser watches a largely or completely blank screen. If the ten small files could be downloaded first, the browsing experience for the person can be much improved because he/she could then have something to look at while waiting for the four large files to download.

The present invention addresses this concern without the need to modify the browser software in any way. In accordance with the invention, the order in which supplemental files referenced in a Web page are downloaded from the server to the requesting client is specified by the designer of the HTML code of the Web page and controlled at the server side regardless of the order in which the client-side Web browser encounters and requests the supplemental files. Particularly, each supplemental file referenced in a Web page has a sequence number associated with it.

In a preferred embodiment, the sequence number is provided as an additional attribute of the tag that calls the supplemental file. Since the client Web browser is a standard Web browser, it will have no idea what the sequence attribute is, which is acceptable since browsers generally will ignore any attribute within a tag that it does not understand. However, at the server-side, when the page is requested, the server parses the page before sending it to the requesting client to find the tags for the supplemental files embedded within the page and reads the associated sequence number attributes. It then builds a queue for serving the supplemental files to the client machine, the supplemental files being queued in the order dictated by the sequence numbers.

Thereafter, regardless of the order in which the browser returns requests for the supplemental files, the server will serve the supplemental files in the order dictated by the queue. Existing browsers already are equipped to receive and cache files and associate such cached files with files referenced in an HTML page. Accordingly, the fact that the supplemental files referenced in a Web page may be received in an order different from the order in which the browser requests them is of no consequence. Specification, page 6, lines 8–13. Accordingly, the invention resides entirely at the server side and will work with any Web browser.

### **The Anders Reference**

Anders discloses a method and apparatus for serving Web pages, including

supplemental files, to a requesting client. However, the method and apparatus disclosed in Anders is entirely different than that of the present invention. Unlike the present invention, Anders requires the Web browser software to be modified to function with the invention. See col. 8, lines 2-6 (which describes the need for a Jammer unpacker "on the client"), col. 8, lines 51-54, and col. 12, line 65 – col. 14, line 7 (which describes in detail the software needed at the browser to implement the invention).

Anders' scheme is entirely different than Applicant's. With reference to Anders' Figure 8, the server transmits the requested page to the requesting client in a particular data stream format 190 that includes the data for the main object (the Web page) and the data for the supplemental objects (such as embedded pictures, etc.) in data entries (packets, such as packets 181-189) that are interleaved with each other in an order selected by the developer. More particularly, the data stream 190 comprises a stream header 180 at the beginning of the stream followed by data definition entries and HTML data entries. Each data definition entry, e.g., 181, 182, 185, 187, defines a supplemental object/file present in the Web page data stream. There is one data definition entry per object/file. The HTML data entries are the actual data of the objects/files (including the main file as well as the supplemental files). Each file will typically consist of many HTML data entries that the browser assembles together to render the whole file. The data definition entry that defines any given object/file must precede the first HTML data entry of that file in order for the browser to know what to do with those HTML data entries when it receives them. Col. 7, line 57 - col. 8, line 36.

The basic premise of Anders' invention is that the publisher 210 (Fig. 11) interleaves the data for the entire web page in a way dictated by itself and serves it to the client that way. The browser, upon receiving each data definition entry, creates an entry in an unpacked object cache (UOC). Then, when the browser starts receiving the HTML data entries corresponding to the supplemental file identified by any given data definition entry, it will append that HTML data to the entry it created in its UOC. In Anders, the browser receives the tags identifying the supplemental files in the order dictated by the data stream 190. Accordingly, the browser may be receiving data of a supplemental file before it receives the HTML data entry that contains the reference to that supplemental file. That is not a problem. Particularly, when the browser reaches the reference to the supplemental file that it has already started downloading and caching in its UOC, the UOC simply forwards the cached data to the browser for rendering.

### **Discussion**

The difference between the former rejections and the present rejections is that the Office is now asserting that, contrary to its previous position, "Anders does not explicitly teach indicates an order". Applicant takes this to mean that the Office is asserting that the data in Anders' web page does not indicate an order in which the supplemental files are to be served.

The Office, however, now asserts that "Krishnan teaches indicates an order (see col. 3, line 45 – col. 4, line 9)" and that it would have been obvious "to implement the teachings of Krishnan into the computer system of Anders to have indicates an order because it would have provided specific functions that comprehensible arrangement among the separate elements of the group".

The Office's analysis is flawed. Krishnan (1) does not teach that for which it has be cited because it does not disclose a Web page containing order data indicating the order in which supplemental files are to be disclosed and (2) even if it did teach such order data, the offered motivation for the proposed combination of features is improper.

Specifically, the assertion that "Krishnan teaches indicates an order" presumably is intended to mean that Krishnan teaches that there is some data in his Web page that indicates the order in which the supplemental files are to be served, since, otherwise, Krishnan would not be relevant. However, the cited portion of Krishnan, column 3, line 45 to column 4, line 9, not only contains no mention whatsoever of the order in which the supplemental files within a Web page are served, it does not mention supplemental files within a Web page at all. Rather, column 3, line 45-column 4, line 9 is in the background section of Krishnan and describes in very basic terms how a client machine requests a Web page and a server serves the Web page in response thereto. It does not, nor is it intended to, describe anything but the basic well-known request/response protocol between a client and server on the Internet for serving Web pages.

Thus, not only does Krishnan not contain any disclosure relating to the ordering of serving of supplemental files referenced in a Web page, it does not appear to contain any disclosure about (1) supplemental files in any context or (2) the order of anything.

Thus, Krishnan does not teach that for which it has been cited.

Even if Krishnan did teach a technique for ordering something, the rejection does not describe the proposed combination. Hence, Applicant cannot possibly specifically address the issue of obviousness as it does not know what the Examiner is asserting is obvious. More specifically, and using claim 1 as an example, the previous Office Action asserted (erroneously) that Anders taught "parsing the code comprising the requested page to detect data within the code that indicates an order in which said supplemental files are to be served". In the latest Office Action, that assertion has been replaced with the assertion that Anders teaches "parsing the code comprising the requested page to detect data within the code which said supplemental files are to be served" and "Anders does not explicitly teach indicates an order. However, Krishnan teaches indicates and order. It would have been obvious ... to implement the teachings of Krishnan into the computer system of Anders to have indicates an order because it would have provided specific functions that comprehensible arrangement among the separate elements of a group".

This assertion, however, contains no description of a proposed combination. It completely lacks any discussion of what part of Krishnan is to be placed into what part of Anders.

As such, it is not a proper obviousness rejection. The Office must explain what the proposed combination is in order to present a prima facie case of obviousness. It has not done so in this case. Applicant cannot address a rejection that is incomplete as Applicant does not understand the rejection.

In fact, the deletion of the words "that indicates an order" between the earlier rejection and the first sentence of the present rejection makes that sentence meaningless. The first sentence quoted above from the present Office Action does not make any sense without the deleted words. In fact, it is not even a sentence. Furthermore, the assertion that Krishnan teaches "indicating an order" also is virtually meaningless. An order of what? What is the order? How is the order indicated?

Even further, the motivation asserted by the Office, i.e., "because it would have provided specific functions that comprehensible arrangement among the separate elements of a group", also is essentially meaningless. Specifically, initially, Applicant does not understand the asserted motivation as quoted above as there appears to be numerous clerical, grammatical, and/or typographical errors in it. Nevertheless at least at some level, it appears that the Office is asserting that the motivation has something to do with a desire to have things in order. However, this motivation does not make sense in the context of any of the prior art, including Anders or the present invention. Specifically, both the conventional technique for serving a Web page as well as Anders already provide an order for serving the supplemental files. Thus, the motivation cannot be the desire to place things in order. They already are in order in Anders as



well as in the more conventional prior art. Oddly, of all of the prior art being discussed, it is Krishnan that contains no teaching of a technique for setting an order for serving supplemental files. Yet this seems to be exactly what it has been cited as teaching.

Accordingly, the rejection must fail because (1) Krishnan adds nothing to the previously recited prior art rejection, which has been overcome, (2) the present rejection is incomprehensible, and (3) the present rejection does not even set forth a proposed combination of the two prior art references that can be addressed by Applicant.

Returning to the Anders reference, as previously argued by Applicant in its appeal brief and previous responses to Office Actions, Anders discloses an interesting technique for indicating the order in which the supplemental files are served. However, it is an entirely different technique than the one of the present invention. Thus, actually contrary to the Office's new position in the new rejections, Anders does, in fact, disclose indicating an order for serving the supplemental files. It just is not the technique claimed in the present application. Rather, as noted above, in Anders, there is nothing that resembles the sequence number attribute embedded within the tag referencing the supplemental file. Furthermore, the server does not parse the code being sent to the client to detect the sequence numbers. That, of course, is because there are no sequence numbers in Anders. Rather, Anders' server builds the data stream 190 using a software module that Anders calls the Publisher 210 (see Figure 11). In Anders, the user specifies the order in which supplemental objects/files are downloaded by the browser, but the information dictating the order is not embedded within the main Web

page itself. Rather, the order is determined by an external software module, namely, the Stream Configurator 212 in the Publisher 210. Thus, while Anders' technology does permit the server to dictate the order in which supplemental files are delivered to the browser, it does so in a way that is entirely different than what is claimed in the present application.

Referring to claim 1, Anders does not disclose (1) "parsing the code comprising the requested page to detect data within the code that indicates an order in which said supplemental files are to be served, said order data comprising data other than the order in which said supplemental files appear in said code defining said Web page".

Applicant made this point during earlier prosecution in this case. The Examiner replied, arguing:

Applicants argue that Anders does not disclose, "parsing the code comprising the requested page to detect data within the code that indicates in order in which said supplemental files are to be served". In response to Applicant's argument, the Patent Office maintain the rejection because Anders does teach parsing the code comprising the requested page to detect data within the code that indicates an order in which said supplemental files are to be served as shown in Col. 10, lines 1-16, 37-40, and 52-62. Anders clearly shows streaming configurator parses web page to identify references to objects and their locations within a page, the designer supplies display sequence information.

The above quoted portion of the Office Action itself discloses the error in the Office's analysis of this issue. The Examiner is absolutely correct in her statements that Anders teaches that the Streaming Configurator parses the Web page to identify references to objects and their locations within the page, but that the designer supplies the display sequence information. However, the Examiner does not appreciate that this

is the opposite of what is claimed in claim 1, i.e., that the sequence for retrieving the supplemental files is given by data embedded within the Web page itself. As noted above, in Anders, it is the Streaming Configurator software module that dictates the order (as specified by the user) in which the supplemental files will be downloaded; it is not HTML tags or any other form of data within the Web page that dictates the order.

Thus, claim 1 patentably distinguishes over Anders by reciting the step of “parsing said code defining said Web page to detect order data within the code that indicates an order in which said supplemental files are to be served, said order data comprising data other than the order in which said supplemental files appear in said code defining said Web page”.

Independent claim 9 also distinguishes over Anders. Claim 9 includes the limitation of “second code indicating an order in which said supplemental files are to be rendered, said second code associated with each of said references and comprising an attribute of a tag associated with said supplemental file”. Hence, claim 9 recites similar features as found in claim 1 discussed above, but in language of differing scope. Therefore, claim 9 distinguishes over Anders for at least all of the reasons discussed above in connection with claims 1. However, claim 9 goes further and recites that the order data is an attribute of a tag. There is nothing in Anders remotely resembling this feature since the order is dictated by the Streaming Configurator. The Office asserted that this is disclosed in col. 11, line 7-col. 12, line 44. However, it is quite clear from col. 11, lines 44-47 that the data identifying the display order does not come from HTML

tags within the file. Col. 11, lines 44-47 state "User supplied display sequence information. This provides information to the Interleaver for the order in which to display the objects".

Independent claim 12 also distinguishes over Anders by virtue of reciting "program code for parsing said code defining the Web page to detect said order data". Since, as previously discussed, the sequence information is not in the Web page in Anders, it obviously cannot retrieve that information by parsing the code of the Web page.

Anders does not obtain the order data from inside the web page. Furthermore, the order is given by the manner in which the data is sequenced in the packet. There is no list of the file order. The data itself is simply ordered within the packet in the order desired by the user. This is an entirely different concept. Therefore, Anders also does not meet the limitation of claim 12 of "constructing a queue in a memory comprising a list of supplemental files in order".

Accordingly, independent claim 12 patentably distinguishes over Anders.

Independent claim 19 includes the limitations "code for parsing said code defining a Web page to detect said order data", "code for constructing a queue in a memory, said queue comprising a list of said supplemental files in said order", and "code for serving said supplemental files to said requesting client machine in said order of said queue". In Anders, as previously mentioned, the display order is not found in the Web page and, therefore, these limitations are not met.

Furthermore, claim 19 specifically recites that the queue "compris[es] a list of said supplemental files in said order". Anders does not meet this limitation. In Anders, there is no list of the file order. The order is given by the manner in which the data itself is interleaved in the packet. This is an entirely different concept.

Accordingly, claim 19 patentably distinguishes over Anders for many of the same reasons discussed above in connection with the other independent claims as well as these additional reasons.

All of the dependent claims distinguish over the prior art for at least the reasons set forth above with respect to the independent claims from which they depend. However, the dependent claims also add even further patentably distinguishing recitations.

For instance, claim 7 depends from claims 1 and 2 and adds that the references to the supplemental files "comprise HTML tags, and said order data comprises attributes of said tags". This is similar to the limitation discussed above in connection with independent claim 9. As noted hereinabove, there is nothing in Anders remotely resembling this since the order is dictated by the Streaming Configurator.

Claim 8 depends from claim 7 and further adds that "said order data attributes are not recognizable by said client machine". This is directly contrary to Anders, in which the client machine must be modified in accordance with Anders' technology in order to recognize Anders' data stream.

Claim 11 depends from independent claim 9 and further distinguishes over

Anders by further describing that the tag comprising the sequence number is an HTML tag. Anders, which does not have a sequence number tag at all, obviously cannot teach such limitations.

Claim 17 depends from independent claim 12 and adds "said references to supplemental files comprise HTML tags" and that "said order data comprises attributes of said tags". As discussed above in connection with independent claim 9 (as well as dependent claim 7), these limitations are not found in Anders.

Claim 18 depends from claim 17 and further adds that "said order data attributes are not recognizable by said client machine". This is not found in Anders as discussed above in connection with claim 8.

Claims 21 and 22 depend from claim 19 and recite essentially the same subject matter as previously discussed in connection with dependent claims 7 and 8, respectively. Accordingly, they even further distinguish over the prior art for the same reasons given above in connection with dependent claims 7 and 8.

**APPLICATION NO. 09/823,331**  
**Response to Action dated 12/09/05**

**DOCKET NO. RSW9-2001-0019-US1**

In view of the foregoing amendments and remarks, this application is now in condition for allowance. Applicant respectfully requests the Examiner to issue a Notice of Allowance at the earliest possible date. The Examiner is invited to contact Applicant's undersigned counsel by telephone call in order to further the prosecution of this case in any way.

Respectfully submitted,

Dated: March 14, 2006



Theodore Naccarella  
Registration No. 33,023  
Synnestvedt & Lechner LLP  
2600 Aramark Tower  
1101 Market Street  
Philadelphia, PA 19107  
Telephone: (215) 923-4466  
Facsimile: (215) 923-2189

Attorney for Applicant

S:\IBM\IBM Raleigh RSW\Patents\P24723 USA\PTO\Response to Final OX.doc